An Integrated Framework for Model-Based Distributed Diagnosis and Prognosis

Anibal Bregon[1], Matthew Daigle[2], and Indranil Roychoudhury[3]

[1] *University of Valladolid, Valladolid, Spain*
anibal@infor.uva.es

[2] *NASA Ames Research Center, Moffett Field, CA 94035, USA*
matthew.j.daigle@nasa.gov

[3] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA*
indranil.roychoudhury@nasa.gov

ABSTRACT

Diagnosis and prognosis are necessary tasks for system reconfiguration and fault-adaptive control in complex systems. Diagnosis consists of detection, isolation and identification of faults, while prognosis consists of prediction of the remaining useful life of systems. This paper presents a novel integrated framework for model-based distributed diagnosis and prognosis, where system decomposition is used to enable the diagnosis and prognosis tasks to be performed in a distributed way. We show how different submodels can be automatically constructed to solve the local diagnosis and prognosis problems. We illustrate our approach using a simulated four-wheeled rover for different fault scenarios. Our experiments show that our approach correctly performs distributed fault diagnosis and prognosis in an efficient and robust manner.

1. INTRODUCTION

Systems health monitoring is essential to guaranteeing the safe, efficient, and reliable operation of engineering systems. Integrated systems health management methodologies include fault diagnosis and prognosis mechanisms, where diagnosis involves *detecting* when a fault has occurred, *isolating* the true fault, and *identifying* the true damage to the system; and prognosis involves *predicting* how much useful life remains in the different components, subsystems, or systems given the diagnosed fault conditions. The information on the fault size and its expected impact on system life can be used to initiate recovery and reconfiguration actions that mitigate the fault or extend system life.

A large body of research exists for both model-based diagnosis (Gertler, 1998; Blanke et al., 2006) and prognosis methods (Luo et al., 2008; Saha & Goebel, 2009; Orchard & Vachtsevanos, 2009). However, the integration of diagnosis and prognosis algorithms is seldom studied. In fact, many diagnosis methodologies leave out the fault identification step, which is necessary to perform a prediction from the current system state. Recently, we presented an integrated model-based framework for diagnosis and prognosis of complex systems, in which we made use of a common modeling framework for modeling both the nominal and faulty system behavior (Roychoudhury & Daigle, 2011).

In (Roychoudhury & Daigle, 2011), the nominal system behavior is estimated using an observer built with the nominal model. Faults are detected when a statistically significant deviation between the nominal estimates and the observed measurements is observed (Biswas et al., 2003). Fault isolation compares the observed measurement deviations against predictions of how the measurements would deviate for each possible fault (Mosterman & Biswas, 1999). Fault identification performs joint state-parameter estimation using multiple observers, where, for each fault, the faulty system model is constructed as the nominal model integrated with a hypothesized fault model (Roychoudhury, 2009). The prognosis module uses, for each fault hypothesis, a prediction model based on its faulty system model and the identified fault parameters, to predict the remaining useful life of the system (Daigle, Saha, & Goebel, 2012). However, this integrated solution performs the diagnosis and prognosis task in a centralized fashion, which is prone to single points of failure, and does not scale well as the size of the system increases.

To overcome such problems, in this work, we leverage recent results for distributed diagnosis (Bregon et al., 2011)

Report Documentation Page

1. REPORT DATE **SEP 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE **An Integrated Framework for Model-Based Distributed Diagnosis and Prognosis**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **NASA Ames Research Center,Moffett Field,CA,94035**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA581775. Annual Conference of the Prognostics and Health Management Society (PHM 2012) held in Minneapolis, Minnesota on September 23-27, 2012. Sponsored by the Office of Naval Research (ONR).

14. ABSTRACT

Diagnosis and prognosis are necessary tasks for system reconfiguration and fault-adaptive control in complex systems. Diagnosis consists of detection, isolation and identification of faults, while prognosis consists of prediction of the remaining useful life of systems. This paper presents a novel integrated framework for model-based distributed diagnosis and prognosis, where system decomposition is used to enable the diagnosis and prognosis tasks to be performed in a distributed way. We show how different submodels can be automatically constructed to solve the local diagnosis and prognosis problems. We illustrate our approach using a simulated fourwheeled rover for different fault scenarios. Our experiments show that our approach correctly performs distributed fault diagnosis and prognosis in an efficient and robust manner.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**11**	

and distributed prognosis (Daigle, Bregon, & Roychoudhury, 2012), which make use of structural model decomposition techniques, to provide a systematic approach to distributing the different diagnosis and prognosis steps presented in (Roychoudhury & Daigle, 2011).

Distributed diagnosis is achieved by designing local distributed subsystems based on global diagnosability analysis of the system, thus computing globally correct distributed diagnosis results without the use of a centralized coordinator (Bregon et al., 2011). These local distributed subsystems are then used to construct local event-based distributed diagnosers for distributed fault isolation. Distributed fault identification is achieved by developing independent local state-parameter estimators for each hypothesized fault. Regarding distributed prediction, in (Daigle, Bregon, & Roychoudhury, 2012) we developed an architecture that enables a large prognosis problem to be decomposed into several independent local subproblems from which local results can be merged into a global result.

The main contribution of this paper is an integrated framework for distributed model-based diagnosis and prognosis of single faults based on structural model decomposition. The proposed framework scales well and the resulting subproblems are typically small and easy to solve, resulting in an efficient and scalable distributed solution to the combined diagnosis and prognosis problem. We perform a number of experiments on a simulated four-wheeled rover testbed (Balaban et al., 2011) to demonstrate and evaluate our approach.

The rest of the paper is organized as follows. Section 2 provides the problem formulation for our diagnosis and prognosis framework. Section 3 describes the distributed architecture and Section 4 briefly introduces its different components. Section 5 presents the case study and experimental results. Finally, Section 6 concludes the paper.

2. PROBLEM FORMULATION

The *nominal system model* is represented as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)),$$
$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)),$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[1]

Faults in the system are represented as changes in the above nominal system model. In this work, we only consider single faults occurring as changes in system parameters, $\boldsymbol{\theta}(t)$. We denote a fault, $f \in F$, as a tuple, (θ, g_f), where $\theta \in \boldsymbol{\theta}$

is the *fault parameter*, and g_f denotes the *fault progression function*, which models the way fault f is manifested in parameter θ, i.e.,

$$\dot{\theta}(t) = g_f(t, \mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t), \mathbf{m}_f(t)),$$

where $\mathbf{x}_f(t) = [\mathbf{x}(t), \theta(t)]^T$, $\boldsymbol{\theta}_f(t) = [\boldsymbol{\theta}(t) \backslash \{\theta(t)\},$ $\boldsymbol{\phi}_f(t)]^T$, $\boldsymbol{\phi}_f(t) \in \mathbb{R}^{n_{\phi_f}}$ is a vector of *fault progression parameters*, and $\mathbf{m}_f(t) \in \mathbb{R}^{n_{m_f}}$ is a process noise vector associated with the fault progression function.

To develop our integrated diagnosis and prognosis framework, the *faulty system model* for fault $f = (\theta, g_f)$ is constructed from the nominal system model by including the parameter as a state and augmenting the state equation by including the fault progression function, i.e.,

$$\dot{\mathbf{x}}_f(t) = \mathbf{f}_f(t, \mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t), \mathbf{v}(t)),$$
$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)),$$

where,

$$\mathbf{f}_f(\cdot) = \left[\begin{array}{c} \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \\ g_f(t, \mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t), \mathbf{m}(t)) \end{array} \right] = \left[\begin{array}{c} \dot{\mathbf{x}}(t) \\ \dot{\theta}(t) \end{array} \right]$$

The goal of diagnosis is to: (*i*) detect a change in some $\theta \in \boldsymbol{\theta}$; (*ii*) isolate, under the single fault assumption, the true fault $f \in F$, i.e., both the parameter θ that has changed, and its fault progression function g_f; and (*iii*) identify (i.e. estimate) the fault by computing $p(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t) | \mathbf{y}(0:t))$, where $\mathbf{y}(0:t)$ denotes all measurements observed up to time t.

The goal of prognosis is to determine the end of (useful) life (EOL) of a system, and/or its remaining useful life (RUL). For a given fault, f, using the fault estimate, $p(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t) | \mathbf{y}(0:t))$, a probability distribution of EOL, $p(\text{EOL}_f(t_P) | \mathbf{y}(0:t_P))$, and/or RUL, $p(\text{RUL}_f(t_P) | \mathbf{y}(0:t_P))$ is computed at a given time point t_P (Daigle, Saha, & Goebel, 2012). Since the prognosis problem is stochastic, EOL/RUL are random variables and we represent them by probability distributions. The acceptable behavior of the system is expressed through a set of n_c constraints, $C_{\text{EOL}_f} = \{c_i\}_{i=1}^{n_c}$, where $c_i : \mathbb{R}^{n_{x_f}} \times \mathbb{R}^{n_{\theta_f}} \times \mathbb{R}^{n_u} \to \mathbb{B}$ maps a given point in the joint state-parameter space given the current inputs, $(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t))$, to the Boolean domain $\mathbb{B} \triangleq [0, 1]$, where $c_i(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t)) = 1$ if the constraint is satisfied (Daigle, Saha, & Goebel, 2012). If $c_i(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t)) = 0$, then the constraint is not satisfied, and the behavior of the system is deemed to be unacceptable. These individual constraints are combined into a single *threshold function* $T_{\text{EOL}_f} : \mathbb{R}^{n_{x_f}} \times \mathbb{R}^{n_{\theta_f}} \times \mathbb{R}^{n_u} \to \mathbb{B}$,

---

[1] Here, we use bold typeface to denote vectors, and use n_a to denote the length of a vector $\mathbf{a}$.

defined as

$$T_{\mathrm{EOL}_f}(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t)) =$$
$$\begin{cases} 1, & 0 \in \{c_i(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t))\}_{i=1}^{n_c} \\ 0, & \text{otherwise.} \end{cases}$$

So, EOL_f may be defined as

$$\mathrm{EOL}_f(t_P) \triangleq$$
$$\inf\{t \in \mathbb{R} : t \geq t_P \text{ and } T_{\mathrm{EOL}_f}(\mathbf{x}_f(t), \boldsymbol{\theta}_f(t), \mathbf{u}(t)) = 1\},$$

i.e., EOL is the earliest time point at which the threshold is reached. RUL is expressed given EOL as

$$\mathrm{RUL}_f(t_P) \triangleq \mathrm{EOL}_f(t_P) - t_P.$$

3. DISTRIBUTED ARCHITECTURE

For a large system, both the diagnosis and prognosis problems are correspondingly large. A centralized approach does not scale well, can be computationally expensive, and prone to single points of failure. Therefore, we propose to decompose the *global* integrated diagnosis and prognosis problem into independent *local* subproblems. In this work, we build on the ideas from structural model decomposition (Blanke et al., 2006; Pulido & Alonso-González, 2004) to compute local independent subproblems, which may be solved in parallel, thus providing scalability and efficiency. Model decomposition is not a new concept, and several approaches have been developed for purposes of system identification, estimation, learning, and diagnosis (Staroswiecki & Declerck, 1989; Pulido & Alonso-González, 2004; Williams & Millar, 1998). Structural model decomposition allows decomposing a global model into a set of local submodels for which local diagnosis and prognosis problems can be directly defined. The global model of the system, denoted as $\mathcal{M}$, is defined as follows.

Definition 1 (Model). The model of a system, $\mathcal{M}$, is a tuple $\mathcal{M} = (X, \Theta, U, Y, C)$, where X is the set of state variables of $\mathbf{x}$, Θ is the set of unknown parameters of $\boldsymbol{\theta}$, U is the set of input variables of $\mathbf{u}$, Y is the set of output variables of $\mathbf{y}$, and C is the set of model constraints of $\mathbf{f}$, $\mathbf{h}$, and EOL constraints of C_{EOL_f}.

The basic idea of the model decomposition problem is to decompose the global system model into a set of submodels satisfying given constraints, such that each submodel contains sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. A submodel in our framework is defined as follows.

Definition 2 (Submodel). A submodel $\mathcal{M}_i$ of a system model $\mathcal{M} = (X, \Theta, U, Y, C)$ is a tuple $\mathcal{M}_i = (X_i, \Theta_i, U_i, Y_i, C_i)$, where $X_i \subseteq X$, $\Theta_i \subseteq \Theta$, $U_i \subseteq X \cup U \cup Y$, and $Y_i \subseteq Y$ are the state, parameter, input, and output variables, respectively,

and $C_i \subseteq C$ are the submodel constraints.[2]

Next, we discuss the fundamental ideas of our model decomposition approach and show the constraints needed to obtain the different submodels for distributed diagnosis and prognosis. Then, we propose our integrated approach.

3.1. Model Decomposition for Distributed Diagnosis and Prognosis

Model decomposition in our framework can be accomplished by using some variables (either measured variables or variables for which the values are known) as local inputs, U_i, such that each one of the submodels satisfies a set of constraints and contains the minimum number of equations to compute a given set of outputs, Y_i. As a result, submodels computed this way contain only a small subset of the equations of the model that are decoupled from the rest of the system model equations. In general, any set of variables in the system can be chosen as the local inputs to the submodel, U_i. The choice of U_i and the constraints to fulfill depends on the particular problem to be solved through model decomposition.

The first model decomposition problem computes minimal submodels from the nominal system model. For this model decomposition problem, constraints are that submodels use the global model inputs and some measured values as local inputs, i.e., $U_i \subseteq U \cup (Y - Y_i)$. An algorithm for computing the set of minimal submodels that satisfies these properties is given in (Daigle et al., 2011), which is based on the model decomposition algorithms presented in (Pulido & Alonso-González, 2004; Bregon et al., 2012).

The second model decomposition problem computes submodels for residual generation and fault isolation. For this model decomposition problem, the constraints are that submodels are constructed by merging the minimal submodels, to fulfill global diagnosability conditions as in (Bregon et al., 2011).[3] Once the globally diagnosable subsystems have been designed, the merged submodels are used for distributed residual generation and to compute event-based local diagnosers for fault isolation. These design and diagnoser computation processes are detailed in (Bregon et al., 2011). Because the subsystems are designed to be globally diagnosable, the resulting local diagnosers are independent, and can provide globally correct diagnosis results without a centralized coordinator.

The third model decomposition problem computes submodels for distributed fault identification. For each consistent fault hypothesis f, the joint state-parameter estimators are computed from the minimal submodels of the faulty system model $\mathcal{M}(f)$ with the constraints that $U_i \subseteq U \cup (Y - Y_i)$ with

---

[2] A faulty submodel for a fault f is denoted as $\mathcal{M}_i(f)$.

[3] In this work, a subsystem is globally diagnosable if all faults in the subsystem are distinguishable from every other fault in the system using only local measurements.
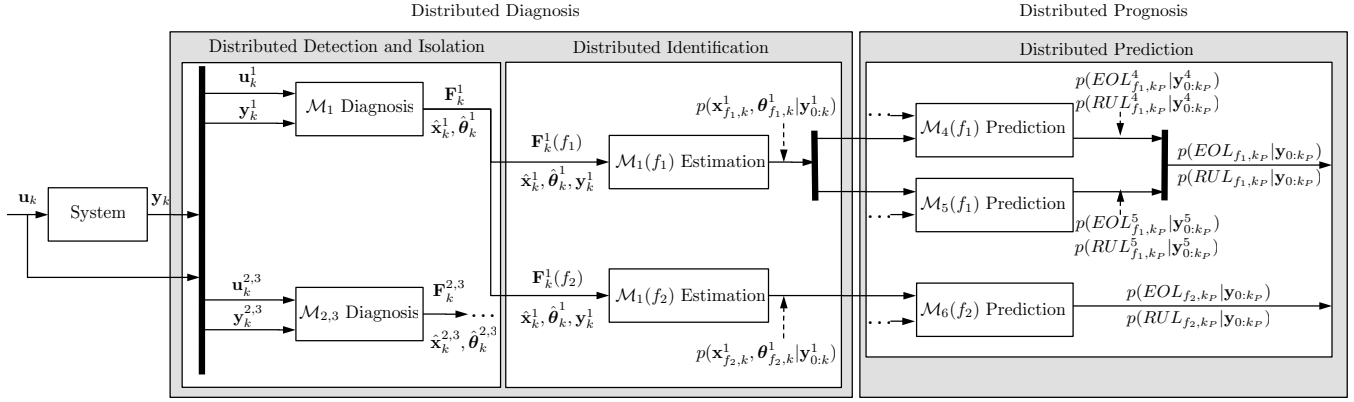
Figure 1. An instantiation of the integrated diagnosis and prognosis architecture.

Y_i as a singleton. It will be shown later that the fault identification module is the central part of our diagnosis-prognosis integration approach and provides the joint state-parameter estimations for the prediction module.

Finally, for distributed prediction, the model decomposition problem starts off from the faulty system model, and, as detailed in (Daigle, Bregon, & Roychoudhury, 2012), it fulfills the following constraints: (*i*) the submodels use $U_i \subseteq U_P$, where $U_P \subseteq X \cup U$ (here, U_P is a set of variables whose future values can be predicted *a priori*, which depends on the hypothesized faults); and (*ii*) each computed submodel has at least one $c \in C_{EOL_f}$ belonging to C_i, and over all submodels, all constraints in C_{EOL_f} are covered. This ensures that T_{EOL_f} may be computed for the system from the local constraints.

3.2. Distributed Architecture

Figure 1 illustrates an example architecture for our distributed diagnosis and prognosis scheme. At each discrete time step, k, the system takes as input both $\mathbf{u}_k$ and $\mathbf{y}_k$ and splits them into local inputs $\mathbf{u}_k^i$ and local outputs $\mathbf{y}_k^i$ for the local diagnosers. Within each $\mathcal{M}_i$ local diagnoser, nominal tracking is performed, computing estimates of nominal measurements, $\hat{\mathbf{y}}_k^i$. The fault detector compares the estimated measurements against the observed measurements, to determine statistically significant deviations for the residual, $\mathbf{r}_k^i = \mathbf{y}_k^i - \hat{\mathbf{y}}_k^i$. Qualitative values of the deviations in the residuals are used by the event-based diagnoser to isolate faults. The set of isolated fault candidates $\mathbf{F}_k^i$ together with the estimated nominal states, $\hat{\mathbf{x}}_k^i$, parameters, $\hat{\boldsymbol{\theta}}_k^i$, and the measurements, $\mathbf{y}_k^i$, are used as input for the corresponding identification module. Identification is performed for each hypothesized fault in a distributed way, e.g., for the isolated faults f_1 and f_2 in Figure 1, we run an instantiation of the identification submodel for each one the faults, i.e., $\mathcal{M}_1(f_1)$ and $\mathcal{M}_1(f_2)$. Fault identification uses the minimal submodels from the faulty system model, and computes local state-parameter es-

timates $p(\mathbf{x}_{f,k}^i, \boldsymbol{\theta}_{f,k}^i | \mathbf{y}_{0:k}^i)$. These local estimates are then used as input to the prediction submodels. In some cases, the local estimates have to be split or merged with other estimates according to the prediction submodels. For example, in the figure, estimates from $\mathcal{M}_1(f_1)$, are used by both local prediction submodels $\mathcal{M}_4(f_1)$ and $\mathcal{M}_5(f_1)$, and those submodels may also need estimates not included within submodel $\mathcal{M}_1(f_1)$. These estimates are typically obtained from the local diagnosers or other fault identification blocks.[4] Distributed prediction modules compute, for each hypothesized fault, local EOL/RUL predictions, $p(EOL_{f,k_P}^i | \mathbf{y}_{0:k_P}^i)$ and $p(RUL_{f,k_P}^i | \mathbf{y}_{0:k_P}^i)$, at given prediction time k_P based on the local EOL constraints. Finally, local predictions are combined into global predictions $p(EOL_{f,k_P} | \mathbf{y}_{0:k_P})$ and $p(RUL_{f,k_P} | \mathbf{y}_{0:k_P})$ for each hypothesized fault. The next section describes the details of the different modules of the distributed integrated diagnosis and prognosis architecture.

4. DIAGNOSIS AND PROGNOSIS APPROACH

Figure 1 shows the basic modules of our distributed integrated approach. In this section we give details on how each module is implemented, and establish the integration between the diagnosis and prognosis tasks.

4.1. Distributed Diagnosis

For distributed diagnosis, each local diagnoser first takes a subset of the local inputs $\mathbf{u}_k^i$ and local outputs $\mathbf{y}_k^i$, to compute an estimate of its output measurements $\hat{\mathbf{y}}_k^i$. Tracking is performed in discrete time using a robust filtering scheme, e.g., the extended or unscented Kalman filter (Julier & Uhlmann, 2004), which provides accurate tracking in the presence of sensor noise, process noise, and discretization error.

---

[4]Since prediction submodels are constructed by using any variable which value can be hypothesized as input, in some cases, prediction submodels cannot always be formed by just merging the minimal estimation submodels. To indicate this, we named our prediction submodels differently from the estimation submodels, e.g., $\mathcal{M}_4(f_1)$ and $\mathcal{M}_5(f_1)$ instead of $\mathcal{M}_1(f_1)$.

For fault detection, a statistical test is used to look for significant deviations in the residual signal $\mathbf{r}_k^i$, which is computed as the difference between $\hat{\mathbf{y}}_k^i$ and the $\mathbf{y}_k^i$. In our approach, we use a Z-test as described in (Biswas et al., 2003).

Fault isolation is performed using local event-based diagnosers, constructed as detailed in the previous section (Bregon et al., 2011; Daigle et al., 2009). Fault isolation is triggered when a fault is detected, and it works as follows. Initially, all event-based local diagnosers start in their initial state, and the set of faulty candidates is empty. Local residual deviations cause the local diagnosers to move from one state to another. These residual deviations are abstracted to a tuple of qualitative symbols (σ_1, σ_2) for each residual signal, where σ_1 represents magnitude changes and σ_2 represents slope changes. A $+$ (resp. $-$) value indicates a change above (resp. below) normal for a measurement residual or a positive (resp. negative) residual slope. A 0 implies no change in the measurement value or a flat residual slope. The symbols are generated using a sliding window technique as described in detail in (Biswas et al., 2003). If there is a match between an event from the current state and a tuple of qualitative symbols generated by any residual, the local diagnoser moves to the next state and remains active. If not, the local diagnoser blocks. This process continues until a local diagnoser reaches an accepting state, which corresponds to a unique isolation result.

In our distributed diagnosis approach, identification submodels, $\mathcal{M}_i(f)$, are obtained, as explained in the previous section, as minimal submodels from the faulty system model. A local state-parameter estimator is constructed for each identification submodel $\mathcal{M}_i(f)$, and produces a local estimate $p(\mathbf{x}_{f,k}^i, \boldsymbol{\theta}_{f,k}^i | \mathbf{y}_{0:k}^i)$ by using an appropriate algorithm. In this paper, we use an unscented Kalman filter (UKF) (Julier & Uhlmann, 2004) with a variance control algorithm (Daigle, Saha, & Goebel, 2012).

4.2. Distributed Prognosis

The local state-parameter estimates for each local distributed prediction module are constructed from the local estimates of the distributed fault identification submodels. Each prediction submodel is made up of a set of states X_i and parameters Θ_i, and constructs a local distribution $p(\mathbf{x}_{f,k}^i, \boldsymbol{\theta}_{f,k}^i | \mathbf{y}_{0:k}^i)$, by assuming that the local state-parameter estimates are sufficiently represented by a mean $\boldsymbol{\mu}^i$ and covariance $\boldsymbol{\Sigma}^i$. For each prediction submodel $\mathcal{M}_i(f)$, we combine the estimates of the local identification submodels that estimate states and parameters in $X_i \cup \Theta_i$ into $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$. If two submodels estimate the same state variable or parameter, then many different techniques can be applied depending on the desired performance of the prediction submodels, e.g., taking the estimate with the smallest variance, or taking an average.

Several approaches can be used to perform prediction for each

---

Algorithm 1 EOL Prediction

Inputs: $\{(\mathbf{x}_{k_P}^{i(j)}, \boldsymbol{\theta}_{k_P}^{i(j)}), w_{k_P}^{i(j)}\}_{j=1}^N$
Outputs: $\{EOL_{k_P}^{i(j)}, w_{k_P}^{i(j)}\}_{j=1}^N$
for $j = 1$ **to** N **do**
  $k \leftarrow k_P$
  $\mathbf{x}_k^{i(j)} \leftarrow \mathbf{x}_{k_P}^{i(j)}$
  $\boldsymbol{\theta}_k^{i(j)} \leftarrow \boldsymbol{\theta}_{k_P}^{i(j)}$
  while $T_{EOL}^i(\mathbf{x}_k^{i(j)}, \boldsymbol{\theta}_k^{i(j)}, \hat{\mathbf{u}}_k^i) = 0$ **do**
    Predict $\hat{\mathbf{u}}_k^i$
    $\boldsymbol{\theta}_{k+1}^{i(j)} \sim p(\boldsymbol{\theta}_{k+1}^i | \boldsymbol{\theta}_k^{i(j)})$
    $\mathbf{x}_{k+1}^{i(j)} \sim p(\mathbf{x}_{k+1}^i | \mathbf{x}_k^{i(j)}, \boldsymbol{\theta}_k^{i(j)}, \hat{\mathbf{u}}_k^i)$
    $k \leftarrow k + 1$
    $\mathbf{x}_k^{i(j)} \leftarrow \mathbf{x}_{k+1}^{i(j)}$
    $\boldsymbol{\theta}_k^{i(j)} \leftarrow \boldsymbol{\theta}_{k+1}^{i(j)}$
  end while
  $EOL_{k_P}^{i(j)} \leftarrow k$
end for

---

prediction submodel. In this work, given the mean and covariance information, we represent the distribution with a set of sigma points derived using the unscented transform. Then, each sigma point is simulated forward to EOL, and we recover the statistics of the EOL distribution given by the sigma points (Daigle & Goebel, 2010).

Algorithm 1 (Daigle, Saha, & Goebel, 2012), shows the pseudocode for the prediction procedure. The algorithm is executed for each submodel i, deriving local EOL predictions using its local threshold function based on the local EOL constraints. For a given submodel, each sample j is propagated forward until $T_{EOL_f}^i(\mathbf{x}_{f,k}^{i(j)}, \boldsymbol{\theta}_{f,k}^{i(j)})$ evaluates to 1. The algorithm hypothesizes future inputs $\hat{\mathbf{u}}_k^i$. Then, the global EOL/RUL is determined by the minimum of the local EOL/RUL distributions for each prediction submodel, i.e., $p(EOL_{f,k_P}^i | \mathbf{y}_{0:k_P}^i)$ and $p(RUL_{f,k_P}^i | \mathbf{y}_{0:k_P}^i)$. To compute this, we sample from each local EOL distribution and take the minimum of the local samples. This is repeated many times and the statistics of the global EOL distribution are computed (Daigle, Bregon, & Roychoudhury, 2012).

5. CASE STUDY

In this section, we apply our distributed diagnosis and prognosis approach to a four-wheeled rover testbed developed at NASA Ames Research Center. We develop a model of the rover, and demonstrate the approach using simulated scenarios.

5.1. Nominal System Modeling

The rover model was originally presented in (Balaban et al., 2011). In this section we summarize the main features and include some extensions to the model.

The rover consists of a symmetric rigid frame with four

independently-driven wheels. The wheel speeds are governed by

$$\dot{\omega}_{FL} = \frac{1}{J_{FL}}\left(\tau_{mFL} - \tau_{fFL} - \tau_{glFL} + \tau_{grFL}\right), \qquad (c_1)$$

$$\dot{\omega}_{FR} = \frac{1}{J_{FR}}\left(\tau_{mFR} - \tau_{fFR} - \tau_{glFR} - \tau_{grFR}\right), \qquad (c_2)$$

$$\dot{\omega}_{BL} = \frac{1}{J_{BL}}\left(\tau_{mBL} - \tau_{fBL} - \tau_{glBL} + \tau_{grBL}\right), \qquad (c_3)$$

$$\dot{\omega}_{BR} = \frac{1}{J_{BR}}\left(\tau_{mBR} - \tau_{fBR} - \tau_{glFR} - \tau_{grBR}\right). \qquad (c_4)$$

The F, B, L, and R subscripts stand for *front*, *left*, *back*, and *right*, respectively. Here, for wheel w, J_w denotes the wheel inertia; τ_{mw} is the motor torque; $\tau_{fw} = \mu_{fw}\omega_w$ is the wheel friction torque, where μ_{fw} is a friction coefficient; $\tau_{glw} = r_w\mu_{gl}(v_w - v)$ is the torque due to slippage, where r_w is the wheel radius, μ_{gl} is a friction coefficient, v_w is the translational wheel velocity, and v is the translation velocity of the rover body; and $\tau_{grw} = r_w\mu_{grw}\omega\cos\gamma$ is the torque due to the rotational movement of the rover body, where μ_{grw} is a friction coefficient, ω is the rotational velocity of the rover body, and $\gamma = \arctan l/b$ with l being the rover's length and b being its width.

The translational velocity v of the rover is described by

$$\dot{v} = \frac{1}{m}\left(F_{glFL} + F_{glFR} + F_{glBL} + F_{glBR}\right), \qquad (c_5)$$

where m is the rover mass, and for wheel w, $F_{glw} = \mu_{gl}(v_w - v)$ is the force due to slippage. The rotational velocity ω is described by

$$\dot{\omega} = \frac{1}{J}(d\cos\gamma F_{glFR} + d\cos\gamma F_{glBR} - d\cos\gamma F_{glFL}$$
$$- d\cos\gamma F_{glBL} - dF_{grFL} - dF_{grFR} - dF_{grBL}$$
$$- dF_{grBR}). \qquad (c_6)$$

Here, J is the rotational inertia of the rover and d is the distance from the center of the rover to each wheel.

The wheels are driven by DC motors with PI control that sets the voltages V applied to the motors. The motor currents i are governed by

$$\dot{i}_{FL} = \frac{1}{L}(V_{FL} - i_{FL}R_{FL} - k_\omega\omega_{FL}), \qquad (c_7)$$

$$\dot{i}_{FR} = \frac{1}{L}(V_{FR} - i_{FR}R_{FR} - k_\omega\omega_{FR}), \qquad (c_8)$$

$$\dot{i}_{BL} = \frac{1}{L}(V_{BL} - i_{BL}R_{BL} - k_\omega\omega_{BL}), \qquad (c_9)$$

$$\dot{i}_{BR} = \frac{1}{L}(V_{BR} - i_{BR}R_{BR} - k_\omega\omega_{BR}), \qquad (c_{10})$$

where L is the motor inductance, R is the motor resistance, and k_ω is an energy transformation term. The motor torque is $\tau_{mw} = k_\tau i_w$, where k_τ is an energy transformation gain. The voltages applied to the motors are determined by the controllers, where for wheel w, $V_w = P * (u_w - \omega_w) + I * e_{iw}$, where P is a proportional gain, u_w is the commanded wheel speed, I is an integral gain, and e_{iw} is the integral error term.

The integral error terms are governed by

$$\dot{e}_{iFL} = u_{FL} - \omega_{FL}, \qquad (c_{11})$$
$$\dot{e}_{iFR} = u_{FR} - \omega_{FR}, \qquad (c_{12})$$
$$\dot{e}_{iBL} = u_{BL} - \omega_{BL}, \qquad (c_{13})$$
$$\dot{e}_{iBR} = u_{BR} - \omega_{BR}. \qquad (c_{14})$$

The batteries, which are connected in series, are described by a simple electrical circuit equivalent model that includes a large capacitance C_b in parallel with a resistance R_p, together in series with another resistance R_s.[5] The battery charge variables q_i are governed by

$$\dot{q}_1 = -V_1/R_{p1} - (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{15})$$
$$\dot{q}_2 = -V_2/R_{p2} - (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{16})$$
$$\dot{q}_3 = -V_3/R_{p3} - (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{17})$$
$$\dot{q}_4 = -V_4/R_{p4} - (i_{FL} + i_{FR} + i_{BR} + i_{BL}). \qquad (c_{18})$$

The available sensors measure the voltages of the batteries,

$$V_1^* = q_1/C_{b1} - R_{s1} * (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{19})$$
$$V_2^* = q_2/C_{b2} - R_{s2} * (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{20})$$
$$V_3^* = q_3/C_{b3} - R_{s3} * (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{21})$$
$$V_4^* = q_4/C_{b4} - R_{s4} * (i_{FL} + i_{FR} + i_{BR} + i_{BL}), \qquad (c_{22})$$

the motor currents,

$$i_{FL}^* = i_{FL}, \qquad (c_{23})$$
$$i_{FR}^* = i_{FR}, \qquad (c_{24})$$
$$i_{BL}^* = i_{BL}, \qquad (c_{25})$$
$$i_{BR}^* = i_{BR}, \qquad (c_{26})$$

and the wheel speeds,

$$\omega_{FL}^* = \omega_{FL}, \qquad (c_{27})$$
$$\omega_{FR}^* = \omega_{FR}, \qquad (c_{28})$$
$$\omega_{BL}^* = \omega_{BL}, \qquad (c_{29})$$
$$\omega_{BR}^* = \omega_{BR}. \qquad (c_{30})$$

Here, the * superscript indicates a measured value.

5.2. Faulty System Modeling

In this work, we consider different faults in the motors and the batteries. First, we consider friction-based damage progression in the motors, resulting in an increase in motor friction over time. For wheel w, the fault progression function is defined as:

$$\dot{\mu}_{fFL} = \nu_{fFL}\,\mu_{fFL}\,\omega_{FL}^2, \qquad (c_{31})$$
$$\dot{\mu}_{fFR} = \nu_{fFR}\,\mu_{fFR}\,\omega_{FR}^2, \qquad (c_{32})$$
$$\dot{\mu}_{fBL} = \nu_{fBL}\,\mu_{fBL}\,\omega_{BL}^2, \qquad (c_{33})$$
$$\dot{\mu}_{fBR} = \nu_{fBR}\,\mu_{fBR}\,\omega_{BR}^2, \qquad (c_{34})$$

---

[5]We use a simple model here only for demonstration purposes. More detailed battery models for prognosis can be found in the literature, e.g., (Saha & Goebel, 2009).

Submodel	X_i	Θ_i	U_i	Y_i	C_i
$\mathcal{M}_1$	q_1	C_{b1}	$i^*_{FL}, i^*_{FR}, i^*_{BL}, i^*_{BR}$	V^*_1	$c_{15}, c_{19}, c_{23}, c_{24}, c_{25}, c_{26}$
$\mathcal{M}_2$	q_2	C_{b2}	$i^*_{FL}, i^*_{FR}, i^*_{BL}, i^*_{BR}$	V^*_2	$c_{16}, c_{20}, c_{23}, c_{24}, c_{25}, c_{26}$
$\mathcal{M}_3$	q_3	C_{b3}	$i^*_{FL}, i^*_{FR}, i^*_{BL}, i^*_{BR}$	V^*_3	$c_{17}, c_{21}, c_{23}, c_{24}, c_{25}, c_{26}$
$\mathcal{M}_4$	q_4	C_{b4}	$i^*_{FL}, i^*_{FR}, i^*_{BL}, i^*_{BR}$	V^*_4	$c_{18}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}$
$\mathcal{M}_5$	i_{FL}, e_{iFL}	R_{FL}	u_{FL}, ω^*_{FL}	i^*_{FL}	$c_7, c_{11}, c_{23}, c_{27}$
$\mathcal{M}_6$	i_{FR}, e_{iFR}	R_{FR}	u_{FR}, ω^*_{FR}	i^*_{FR}	$c_8, c_{12}, c_{24}, c_{28}$
$\mathcal{M}_7$	i_{BL}, e_{iBL}	R_{BL}	u_{BL}, ω^*_{BL}	i^*_{BL}	$c_9, c_{13}, c_{25}, c_{29}$
$\mathcal{M}_8$	i_{BR}, e_{iBR}	R_{BR}	u_{BR}, ω^*_{BR}	i^*_{BR}	$c_{10}, c_{14}, c_{26}, c_{30}$
$\mathcal{M}_9$	$\omega_{FL}, v, \omega, \mu_{fFL}$	ν_{fFL}	$i^*_{FL}, \omega^*_{FR}, \omega^*_{BL}, \omega^*_{BR}$	ω^*_{FL}	$c_1, c_{31}, c_5, c_6, c_{23}, c_{28}, c_{29}, c_{30}$
$\mathcal{M}_{10}$	$\omega_{FR}, v, \omega, \mu_{fFR}$	ν_{fFR}	$i^*_{FR}, \omega^*_{FL}, \omega^*_{BL}, \omega^*_{BR}$	ω^*_{FR}	$c_2, c_{32}, c_5, c_6, c_{24}, c_{27}, c_{29}, c_{30}$
$\mathcal{M}_{11}$	$\omega_{BL}, v, \omega, \mu_{fBL}$	ν_{fBL}	$i^*_{BL}, \omega^*_{FL}, \omega^*_{FR}, \omega^*_{BR}$	ω^*_{BL}	$c_3, c_{33}, c_5, c_6, c_{25}, c_{27}, c_{28}, c_{30}$
$\mathcal{M}_{12}$	$\omega_{BR}, v, \omega, \mu_{fBR}$	ν_{fBR}	$i^*_{BR}, \omega^*_{FL}, \omega^*_{FR}, \omega^*_{BL}$	ω^*_{BR}	$c_4, c_{34}, c_5, c_6, c_{26}, c_{27}, c_{28}, c_{29}$

Table 1. Set of minimal submodels for the rover testbed computed from the nominal system model.

Submodel	X_i	Θ_i	U_i	Y_i	C_i
$\mathcal{M}_{5,9}$	$\omega_{FL}, v, \omega, \mu_{fFL}, i_{FL}, e_{iFL}$	ν_{fFL}, R_{FL}	$u_{FL}, \omega^*_{FR}, \omega^*_{BL}, \omega^*_{BR}$	ω^*_{FL}, i^*_{FL}	$C_5 \cup C_9$
$\mathcal{M}_{6,10}$	$\omega_{FR}, v, \omega, \mu_{fFR}, i_{FR}, e_{iFR}$	ν_{fFR}, R_{FR}	$u_{FR}, \omega^*_{FL}, \omega^*_{BL}, \omega^*_{BR}$	ω^*_{FR}, i^*_{FR}	$C_6 \cup C_{10}$
$\mathcal{M}_{7,11}$	$\omega_{BL}, v, \omega, \mu_{fBL}, i_{BL}, e_{iBL}$	ν_{fBL}, R_{BL}	$u_{BL}, \omega^*_{FL}, \omega^*_{FR}, \omega^*_{BR}$	ω^*_{BL}, i^*_{BL}	$C_7 \cup C_{11}$
$\mathcal{M}_{8,12}$	$\omega_{BR}, v, \omega, \mu_{fBR}, i_{BR}, e_{iBR}$	ν_{fBR}, R_{BR}	$u_{BR}, \omega^*_{FL}, \omega^*_{FR}, \omega^*_{BL}$	ω^*_{BR}, i^*_{BR}	$C_8 \cup C_{12}$
$\mathcal{M}_{1,2,3,4}$	q_1, q_2, q_3, q_4	$C_{b1}, C_{b2}, C_{b3}, C_{b4}$	$i^*_{FL}, i^*_{FR}, i^*_{BL}, i^*_{BR}$	$V^*_1, V^*_2, V^*_3, V^*_4$	$C_1 \cup C_2 \cup C_3 \cup C_4$

Table 2. Residual generation and fault isolation submodels.

where μ_{fw} is the fault parameter, and ν_{fw} is the fault progression parameter.

We also consider abrupt resistance increases in the motors, represented as an abrupt change in parameter R_w for wheel w, with ΔR_w as the fault progression parameter.

For the batteries, we consider abrupt capacitance decreases, represented as an abrupt change in parameter C_{bi} for capacity i. ΔC_{bi} is the fault progression parameter.

We are interested in predicting when any of the rover batteries are at their charge threshold, beyond which the batteries will be damaged. These faults can cause the charge thresholds to be reached earlier since they will affect current draw. The constraints are given as

$$q_1 > q^-, \qquad (c_{35})$$
$$q_2 > q^-, \qquad (c_{36})$$
$$q_3 > q^-, \qquad (c_{37})$$
$$q_4 > q^-, \qquad (c_{38})$$

where the charge threshold is given by $q^- = 2 \times 10^4$ C. The rover cannot be operated when any of the constraints c_{35}–c_{38} are violated.

5.3. Results

To demonstrate the validity of the approach, we describe two different faulty scenarios of the rover. In the first, friction damage is progressing on one motor, and in the second, a capacitance decrease occurs in one battery. In all cases, the rover travels between various waypoints, moving at an average speed of 0.5 m/s. Table 1 shows the minimal submodels for the rover derived by using measured values as local inputs. Table 2 shows the submodels for residual generation

Submodel	X_i	Θ_i	U_i	Y_i	C_i
$\mathcal{M}_{17}(C_{b1})$	q_1, C_{b1}	ΔC_{b1}	$i_{FL}, i_{FR}, i_{BL}, i_{BR}$	$\varnothing$	c_{15}, c_{19}, c_{35}
$\mathcal{M}_{18}(C_{b2})$	q_2, C_{b2}	ΔC_{b2}	$i_{FL}, i_{FR}, i_{BL}, i_{BR}$	$\varnothing$	c_{16}, c_{20}, c_{36}
$\mathcal{M}_{19}(C_{b3})$	q_3, C_{b3}	ΔC_{b3}	$i_{FL}, i_{FR}, i_{BL}, i_{BR}$	$\varnothing$	c_{17}, c_{21}, c_{37}
$\mathcal{M}_{20}(C_{b4})$	q_4, C_{b4}	ΔC_{b4}	$i_{FL}, i_{FR}, i_{BL}, i_{BR}$	$\varnothing$	c_{18}, c_{22}, c_{38}

Table 4. Prediction submodels for capacitance faults.

and fault isolation. These submodels have been designed to obtain globally diagnosable subsystems by using the design algorithm in (Bregon et al., 2011). In this work, we have considered five subsystems, one for each wheel components and another one for the batteries. For example, the subsystem for the front left wheel components is not globally diagnosable if we only consider submodel $\mathcal{M}_9$ (which includes the front left wheel friction wear parameter, ν_{fFL}). The design algorithm in (Bregon et al., 2011) determines that we need to merge submodels $\mathcal{M}_5$ and $\mathcal{M}_9$ to make the front left wheel subsystem globally diagnosable. The process is similar for the rest of the subsystems.
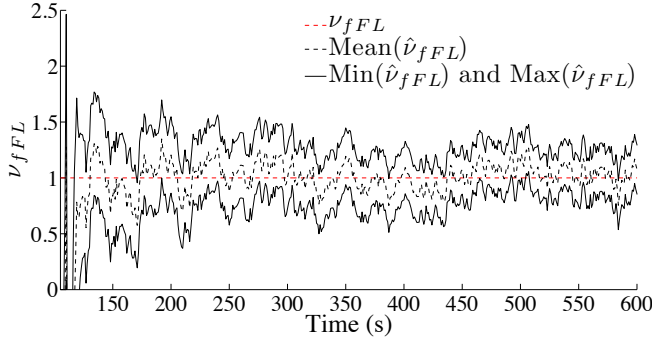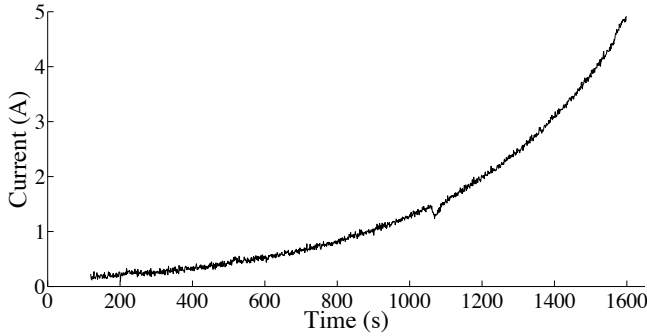
Minimal submodels for identification can be computed from the minimal submodels in Table 1 by defining the fault progression function (if necessary), and by making the fault parameter to become a state and the corresponding fault progression parameter to become the parameter. Regarding prediction, the correct prediction submodels to use depend on the scenario, as will be shown later.

5.3.1. Friction Damage Progression

We first consider a scenario in which, for the front-left motor, the friction begins to increase. The friction damage progression begins at $t = 50$ s with friction wear parameter $\nu_{fFL} = 1 \times 10^{-3}$ s. A fault is detected by the local di-

Submodel	X_i	Θ_i	U_i	Y_i	C_i
$\mathcal{M}_{13}(\mu_{fFL})$	$q_1,i_{FL},e_{iFL},\omega_{FL},\mu_{fFL}$	C_{b1},ν_{fFL},R_{FL}	$u_{FL},v,\omega,i_{FR},i_{BL},i_{BR}$	$\varnothing$	$c_1,c_{31},c_7,c_{11},c_{15},c_{35}$
$\mathcal{M}_{14}(\mu_{fFL})$	$q_2,i_{FL},e_{iFL},\omega_{FL},\mu_{fFL}$	C_{b2},ν_{fFL},R_{FL}	$u_{FL},v,\omega,i_{FR},i_{BL},i_{BR}$	$\varnothing$	$c_1,c_{31},c_7,c_{11},c_{16},c_{36}$
$\mathcal{M}_{15}(\mu_{fFL})$	$q_3,i_{FL},e_{iFL},\omega_{FL},\mu_{fFL}$	C_{b3},ν_{fFL},R_{FL}	$u_{FL},v,\omega,i_{FR},i_{BL},i_{BR}$	$\varnothing$	$c_1,c_{31},c_7,c_{11},c_{17},c_{37}$
$\mathcal{M}_{16}(\mu_{fFL})$	$q_4,i_{FL},e_{iFL},\omega_{FL},\mu_{fFL}$	C_{b4},ν_{fFL},R_{FL}	$u_{FL},v,\omega,i_{FR},i_{BL},i_{BR}$	$\varnothing$	$c_1,c_{31},c_7,c_{11},c_{18},c_{38}$

Table 3. Prediction submodels using commanded wheel speeds and rover velocities as local inputs.

Figure 2. Estimated ν_{fFL} values.

Figure 3. Current i_{FL}^* increase through time.

agnoser computed from submodel $\mathcal{M}_{5,9}$ at 119.25 s, via an increase in the motor current i_{FL}. The initial candidate list is immediately reduced to one candidate, $\{\nu_{fFL}\}$, based on the signatures and orderings (other faults in the front left wheel, like R_{FL}, produce different fault signatures). Thus the true fault is isolated.

Fault identification is initiated once the candidate is isolated. For the friction damage progression fault, the wear rate ν_{fFL} estimate averages to $\nu_{fFL} = 1 \times 10^{-3}$ s with very small output error. Figure 2 shows the wear parameter estimate for friction damage.

As a result of the continuously increasing friction, the current drawn by the motor increases as well in order for the motor controller to maintain the same desired wheel speed (Figure 3 shows this increase in the current through time). Hence, the total current drawn from the batteries is increased, and EOL occurs around half an hour. Because i_{FL} is constantly changing, and in a way that is dependent on the motor state, it is

incorrect to use it as a local input for prediction and to decompose the prediction problem into independent local prediction problems for the batteries and motors, i.e., it is not known a priori. Therefore, we compute submodels using as local inputs average values for the remaining motor currents, average commanded wheel speeds, and average rover translational velocity v and rotational velocity ω. The prediction submodels for this case are shown in Table 3. EOL for this fault is computed by merging the local EOL from those submodels in the table. Note that the prediction submodels used in this case do not correspond directly to those used for estimation. So, when constructing the estimate for $\mathcal{M}_{13}$, for example, it takes the estimates from $\mathcal{M}_1$ and $\mathcal{M}_9$.

The prediction results are shown in Figure 4. The increased friction causes the batteries to discharge faster, and EOL occurs around 1650 s. Here, we used the relative accuracy (RA) as a measure of prediction accuracy, and the relative standard deviation (RSD) as a measure of spread. Each prediction metric is averaged over multiple prediction points (one every 100 s of usage) (see (Saxena et al., 2010; Daigle, Saha, & Goebel, 2012) for the mathematical definitions of these metrics). For this experiment, RA averages to 91.63% and RSD averages to 16.26%.

For the sake of comparison, we also ran this experiment using the centralized approach. Figure 5 shows the prediction results obtained. Looking at the prediction metrics, we see that the centralized approach behaved very similar to the distributed approach but a little bit worse, with RA averaging 90.90% and RSD averaging 17.72%. However, this is just a particular example, but, in general, both approaches obtain equivalent results.

5.3.2. Capacitance Decrease

As a second scenario, we consider a capacitance decrease fault in battery 3 of the rover, C_{b3}. The fault begins at $t = 50$ s with an abrupt decrease from 2000 to 1800 in the capacity of the battery. The fault is detected immediately by the local diagnoser computed from submodel $\mathcal{M}_{1,2,3,4}$ at 50.0 s, via an increase in the voltage V_3. The fault candidate is immediately isolated, $\{C_{b3}\}$, based on the signatures and orderings, thus starting the fault identification. For the capacitance fault, the estimated value of the capacitance averaged $C_{b3} = 1798.6$ C with very small output error. As a result of the decrease in capacitance, the battery discharges at a faster rate, and so reaches end of discharge more quickly. The prediction sub-
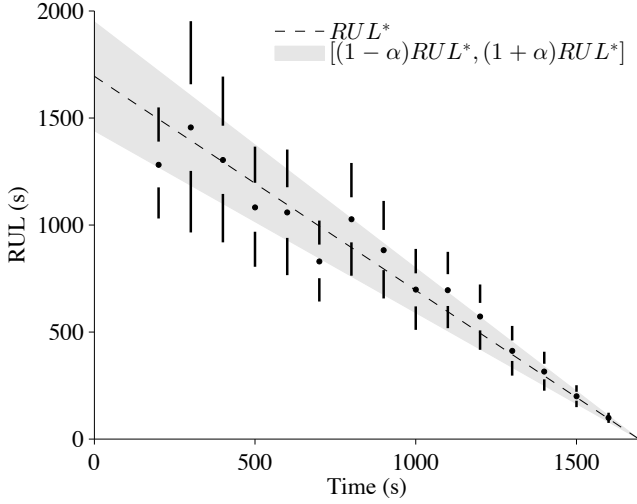
Figure 4. Predicted RUL of the rover for the distributed approach. The mean is indicated with a dot and confidence intervals for 5% and 95% by lines. The gray cone depicts an accuracy requirement of 15%.
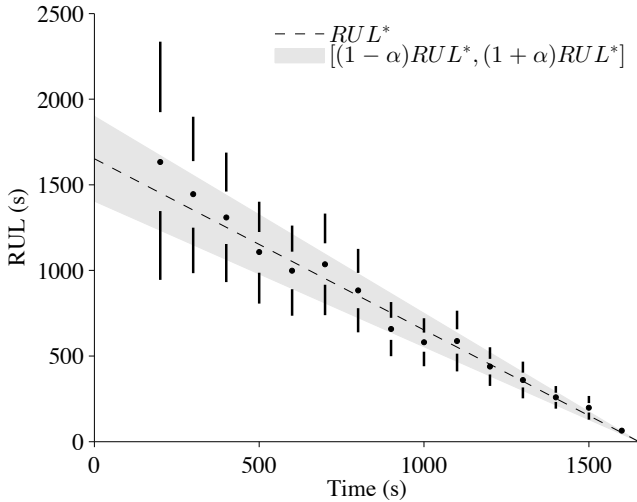

Figure 5. Predicted RUL of the rover for the centralized approach. The mean is indicated with a dot and confidence intervals for 5% and 95% by lines. The gray cone depicts an accuracy requirement of 15%.

models for faults in the capacity of the batteries are shown in Table 4. For this scenario, with a fault in C_{b3}, we used submodel $\mathcal{M}_{19}$, obtaining RA average to 98.25% and RSD average to 10.12%.

6. CONCLUSIONS

This paper presented a distributed integrated model-based diagnosis and prognosis framework. Our approach starts off with a common modeling paradigm to model both the nominal behavior and fault progression, and then proposes a framework where the global system model is decomposed into smaller independent submodels. These submodels are then used to distribute the different diagnosis and prognosis tasks. Model decompostion is carried out based on the requirements and constraints of each task. We demonstrated our approach on a four-wheeled rover testbed, where we diagnosed faults and prognosed the EOL/RUL accurately. We compared results obtained by using our distributed approach against those obtained using a centralized approach, showing that both approaches obtain the same results.

Most approaches in the literature focus in either the diagnosis or the prognosis task. Some works have proposed the integration of both tasks within a common framework (Patrick et al., 2007; Orchard & Vachtsevanos, 2009; Roychoudhury & Daigle, 2011), however, unlike our approach, these approaches perform the diagnosis and prognosis tasks in a centralized way, thus suffering from scalability issues due to the large number of states and parameters in real-world systems. To the best of our knowledge, there is no approach in the literature which combines, in a distributed way, the diagnosis and prognosis tasks. Our approach is limited by the number and location of the sensors in the system. Since our decomposition algorithm is guided by the set of available sensors, the distribution capabilities of the approach is determined by them.

In future, we will apply this approach to larger systems, to study the scalability of our diagnosis and prognosis scheme, and will perform a more detailed comparison against the results obtained by using a centralized approach. We will also extend the capability of this approach to hybrid systems, as well as diagnosis and prognosis of multiple faults.

REFERENCES

Balaban, E., Narasimhan, S., Daigle, M., Celaya, J., Roychoudhury, I., Saha, B., et al. (2011, September). A Mobile Robot Testbed for Prognostics-Enabled Autonomous Decision Making. In *Annual Conference of the Prognostics and Health Management Society* (p. 15-30). Montreal, Canada.

Biswas, G., Simon, G., Mahadevan, N., Narasimhan, S., Ramirez, J., & Karsai, G. (2003, June). A robust method for hybrid diagnosis of complex systems. In *Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes* (pp. 1125–1131).

Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*. Springer.

Bregon, A., Biswas, G., & Pulido, B. (2012). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, *42*(3), 751-763.

Bregon, A., Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., & Pulido, B. (2011, Oct). Improving Distributed Diagnosis Through Structural Model Decomposition. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis* (p. 195-202). Murnau, Germany.

Daigle, M., Bregon, A., & Roychoudhury, I. (2011, September). Distributed Damage Estimation for Prognostics Based on Structural Model Decomposition. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011* (p. 198-208).

Daigle, M., Bregon, A., & Roychoudhury, I. (2012). *Distributed prognostics based on structural model decomposition.* (Manuscript submitted for publication)

Daigle, M., & Goebel, K. (2010, October). Improving Computational Efficiency of Prediction in Model-based Prognostics Using the Unscented Transform. In *Annual Conf. of the Prognostics and Health Management Society 2010.*

Daigle, M., Koutsoukos, X., & Biswas, G. (2009, July). A Qualitative Event-based Approach to Continuous Systems Diagnosis. *IEEE Trans. on Control Systems Technology*, *17*(4), 780–793.

Daigle, M., Saha, B., & Goebel, K. (2012, March). A comparison of filter-based approaches for model-based prognostics. In *Proceedings of the 2012 IEEE Aerospace Conference.*

Gertler, J. J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. New York, NY: Marcel Dekker, Inc.

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. *Proc. of the IEEE*, *92*(3), 401–422.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-based prognostic techniques applied to a suspension system. *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Mosterman, P. J., & Biswas, G. (1999, November). Diagnosis of Continuous Valued Systems in Transient Operating Regions. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Trans. on*, *29*(6), 554 - 565.

Orchard, M. E., & Vachtsevanos, G. (2009). A Particle-filtering Approach for On-line Fault Diagnosis and Failure Prognosis. *Trans. of the Institute of Measurement and Control*, *31*(3/4), 221-246.

Patrick, R., Orchard, M. E., Zhang, B., Koelemay, M., Kacprzynski, G., Ferri, A., et al. (2007, September). An Integrated Approach to Helicopter Planetary Gear Fault Diagnosis and Failure Prognosis. In *Proc. of the 42nd Annual Systems Readiness Technology Conf.* Baltimore, MD, USA.

Pulido, B., & Alonso-González, C. (2004). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part B, Special Issue on Diagnosis of Complex Systems*, *34*(5), 2192-2206.

Roychoudhury, I. (2009). *Distributed Diagnosis of Continuous Systems: Global Diagnosis Through Local Analysis*. Unpublished doctoral dissertation, Vanderbilt University.

Roychoudhury, I., & Daigle, M. (2011, October). An integrated model-based diagnostic and prognostic framework. In *Proceedings of the 22nd International Workshop on Principles of Diagnosis* (p. 44-51).

Saha, B., & Goebel, K. (2009, September). Modeling Li-ion battery capacity depletion in a particle filtering framework. In *Proc. of the Annual Conf. of the Prognostics and Health Management Society 2009.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *Int. Journal of Prognostics and Health Management*.

Staroswiecki, M., & Declerck, P. (1989, July). Analytical redundancy in nonlinear interconnected systems by means of structural analysis. In *IFAC Symp. on Advanced Information Processing in Automatic Control.*

Williams, B., & Millar, B. (1998). Decompositional Model-based Learning and its analogy to diagnosis. In *Proc. of the Fifteenth National Conf. on Artificial Intelligence, AAAI'98* (p. 197-204).

BIOGRAPHIES

Anibal Bregon received his B.S., M.S., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant

with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid.

Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis of complex physical systems.

Matthew Daigle received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively.

From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Sci-

entist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

Dr. Daigle is a member of the Prognostics and Health Management Society and the IEEE. He is a recipient of a University Graduate Fellowship from Vanderbilt University, a best paper award in the Annual Conference of the Prognostics and Health Management Society 2011, a NASA Ames Group Achievement Award in 2011, and an Ames Contractor Council Excellence Award in 2011. He has published over 40 peer-reviewed papers in the area of Systems Health Management.

Indranil Roychoudhury received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist.

Dr. Roychoudhury is a member of the Prognostics and Health Management Society and the IEEE. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems.